Fusion Engineering and Design 192 (2023) 113606

Contents lists available at ScienceDirect

Fusion Engineering and Design

journal homepage: www.elsevier.com/locate/fusengdes

# Fast model-based scenario optimization in NSTX-U enabled by analytic gradient computation

Brian R. Leard [*], Sai Tej Paruchuri, Tariq Rafiq, Eugenio Schuster

*Mechanical Engineering and Engineering Mechanics, Lehigh University, Bethlehem, PA, USA*

ARTICLE INFO

*Keywords:*
Feedforward optimization
Scenario planning
Analytic gradients
NSTX-U

ABSTRACT

Model-based optimization offers a systematic approach to advanced scenario planning. In this case, the feedforward-control inputs (actuator trajectories) that are needed to attain and sustain a desired scenario are obtained by solving a nonlinear constrained optimization problem. This class of problems generally minimize a cost function that measures the difference between desired and actual plasma states. Several numerical optimization algorithms, such as sequential quadratic programming, require repeated calculation of the cost function gradients with respect to the input trajectories. Calculating these gradients numerically can be computationally intensive, increasing the time needed to solve the feedforward-control optimization problem. This work introduces a method to analytically calculate these cost function gradients from the current profile evolution model. This can significantly reduce the computational time and allow for fast feedforward-control optimization, which would eventually enable optimal scenario planning between discharges. The performance of the feedforward optimizer with analytical gradients is compared to a traditional optimization algorithm based on numerical gradients for different NSTX-U scenarios. The plasma dynamics in the optimization algorithm are simulated using the Control Oriented Transport SIMulator (COTSIM). Results of the work show that analytical gradients consistently reduce the computation time while achieving trajectories that are comparable to those obtained by traditional optimization algorithms based on numerical gradients.

## 1. Introduction

Advanced tokamak scenarios are often difficult to realize due to the nonlinear and highly-coupled dynamics governing plasma evolution. One traditional method at reaching these scenarios is using an experimental trial-and-error approach, which involves repetitive testing of different combinations of actuator values with the goal of discovering desirable plasma states. However, this method requires many costly and time-consuming tokamak shots, and when testing in unexplored regions of the tokamak operating space there is a risk of causing damage to the machine. Another method is model-based scenario planning via feedforward optimization, which was proposed in [1] and developed for numerous tokamak devices, such as DIII-D [2], EAST [3], NSTX- U [4], and TCV [5].

Feedforward optimization relies on minimizing a cost function that is defined in terms of the error between the actual state and the desired state. One of the most commonly used set of techniques to solve such an optimization problem is gradient-based methods. These methods employ an iterative approach, which involves guessing the inputs, running a simulation, and updating the guess based on the outputs of that simulation. To update the input guess, gradients of the cost function are calculated for each iteration. The next step is taken in the direction that has the greatest reduction of the gradient. Traditionally, these gradients are calculated numerically. Numerical calculation of the gradient requires knowledge of the state evolution generated by different optimized parameters. Thus, multiple feedforward simulations must be carried out to compute the numerical gradients. Depending on the complexity of the simulation, this can be a computationally intensive process and therefore significantly increases the calculation time. This work proposes to calculate these gradients analytically to significantly reduce the computational burden of the feedforward optimization process. Particularly, a feedforward optimizer using sequential quadratic programming (SQP) with analytical gradients has been developed to achieve a desired safety-factor ($q$) profile evolution in NSTX-U scenarios. The evolution of $q$ is modeled by the magnetic diffusion equation, which is complemented by half-dimensional control-oriented models. While this approach has already been followed in some previous work like [5], in this work a comparison between analytical and numerical gradient computation is provided and quantified. The addition of analytical gradients to the solver may allow for feedforward optimization to be utilized in situations that may require fast scenario

[*] Corresponding author.
*E-mail address:* brian.leard@lehigh.edu (B.R. Leard).

https://doi.org/10.1016/j.fusengdes.2023.113606
Received 31 October 2022; Received in revised form 30 January 2023; Accepted 24 February 2023
Available online 10 March 2023

planning, such as in between tokamak discharges. For instance, if there is an unexpected actuator failure during a tokamak run, fast feedforward optimization can be used to update the trajectories of the available actuators to achieve the desired target scenario as closely as possible during the following tokamak run.

The advantage of the open-loop (feedforward) approach to scenario control resides on the capability of dealing with transport models of arbitrary complexity because all the computations are carried out off-line. The disadvantage, on the other hand, resides in its sensitivity to unmodeled plasma dynamics (i.e., model uncertainties) and changing plasma conditions. A closed-loop (feedback) control component must be added to the optimal open-loop (feedforward) control component obtained as solution of the actuator-trajectory optimization problem in order to increase the robustness of the overall controller against initial-condition perturbations, model uncertainties, and disturbances. Therefore, the feedforward-control schemes like the one proposed in this work are paired with feedback controllers [6–8] to successfully regulate multiple plasma properties around values dictated by the desired plasma scenario.

The structure of this paper is as follows. Section 2 overviews the SQP based feedforward optimization method and discusses where the gradient calculation is required. Section 3 covers the simulation model for the current profile dynamics. Section 4 formulates the optimization problem. Section 5 covers how the gradients of the cost function with respect to the optimization parameters (inputs) are derived analytically from the plasma dynamics. Section 6 shows the simulation results in comparison to a feedforward solver with numerically derived gradients.

## 2. Feedforward optimization for scenario planning

Consider a nonlinear dynamic system governed by

$$\dot{x} = f(x, u), \tag{1}$$

$$y = g(x, u), \tag{2}$$

where $x \in \mathbb{R}^{M\times 1}$ is the state of the system, $\dot{x}$ denotes its time derivative, $u \in \mathbb{R}^{N\times 1}$ is the input of the system, and $y \in \mathbb{R}^{P\times 1}$ is the output of the system. The state function is represented by $f(x, u)$, and the output function is represented by $g(x, u)$. The goal of feedforward optimization is to determine the combination of input trajectories $u$ that produce a desired target output $y$. Since the governing equations are defined in continuous time, the associated optimization problem of determining the input trajectories to achieve the desired target is infinite-dimensional. Approximations are required to make computation feasible. In this work, input parameterization is used to reduce the dimensionality of the optimization problem. Parameterization transforms the infinite dimensional problem into a finite dimensional problem by selecting discrete points in time where the inputs will be varied. Any interpolation method can now be used to calculate the input values at all other time points. Linear interpolation was used in this work, however a more sophisticated interpolation method such as fitting a spline function is feasible. These chosen discrete control times are denoted as $\bar{t}_i$, $i = 1, 2, \dots \bar{T}$ and stored in the vector $\bar{\boldsymbol{t}}$. Each input value, $u_n(t)$, $n = 1, 2, \dots, N$, is now transformed,

$$u_n(t) \longrightarrow \boldsymbol{u}_n = \left[u_n(\bar{t}_1), \quad \dots \quad u_n(\bar{t}_{\bar{T}}),\right]' \in R^{\bar{T}\times 1} \tag{3}$$

and the input vector to the optimization problem is now

$$\boldsymbol{u} = \left[\boldsymbol{u}_1, \quad \dots \quad \boldsymbol{u}_n, \quad \dots \quad \boldsymbol{u}_N\right]' \in \mathbb{R}^{(N*\bar{T})\times 1}. \tag{4}$$

By parameterizing the inputs, the optimization problem becomes viable to be solved. A user defined cost function $J(\boldsymbol{u})$ is created to mathematically represent the distance to the target output $y^*$ and encapsulate other potential control objectives. This cost function is minimized by varying the parameterized inputs $\boldsymbol{u}$, while being subject to various state and actuator constraints. The feedforward optimization process can be mathematically expressed as a constrained minimization problem,

$$\min_{\boldsymbol{u}} \; J(y, \boldsymbol{u}, y^*), \tag{5}$$

$$\text{such that: } \dot{x} = f(x, \boldsymbol{u}), \tag{6}$$

$$y = g(x, \boldsymbol{u}), \tag{7}$$

$$\hat{x} \le x \le \bar{x}, \tag{8}$$

$$\hat{\boldsymbol{u}} \le \boldsymbol{u} \le \bar{\boldsymbol{u}}. \tag{9}$$

In the above formulation, (5) is the to-be-minimized cost function, (6) and (7) represent the state dynamics, and (8) and (9) are the state and input constraints. The minimum allowable value of the inputs is denoted by $\hat{\boldsymbol{u}}$ and the maximum value is denoted by $\bar{\boldsymbol{u}}$. Similarly, the state is constrained by the minimum value $\hat{x}$ and the maximum value $\bar{x}$.

This minimization problem can be solved using an iterative method known as Sequential Quadratic Programming (SQP). Each iteration observes the following general sequence. First, an estimate of the optimal input parameters, $\boldsymbol{u}$ is produced. Second, a simulation of the system dynamics is run using these parameters. Third, the cost function is calculated. Fourth, a new estimate of the optimal input parameters with the ultimate goal of minimizing the cost function is determined from the previous sequence. The full mathematical formulation of SQP can be found in [9,10]. What is important to note about this method is that in the fourth step, the new estimate of the optimal input parameters are found using the gradient of the cost function. As mentioned, these gradients are conventionally solved numerically using successive evaluations of the cost function. Depending on the complexity of the state dynamics (6), this can be a major computational burden. In this work, we propose computing the gradients analytically using a control-oriented model, which is presented in the following section.

## 3. Plasma evolution model

### 3.1. Magnetic diffusion equation

In a tokamak, a magnetic flux surface is defined by points having identical poloidal magnetic flux $\Psi$. Under ideal magneto-hydrodynamic conditions, these magnetic flux surfaces are nested and certain plasma properties like the safety factor, plasma pressure, and current density are constant along these surfaces. Additionally, the plasma is assumed to be toroidally axisymmetric. Therefore, a plasma parameter that is constant on each magnetic flux surface can be used to index the flux surface, and can be considered the spatial coordinate for the now one-dimensional problem. In this work, the normalized mean effective minor radius of the flux surface is considered the indexing variable. The mean effective minor radius is defined as $\rho \triangleq \sqrt{\Phi/(B_{\phi,0}\pi)}$, where $\Phi$ is the toroidal magnetic flux and $B_{\phi,0}$ is the vacuum toroidal magnetic field at the major radius of the tokamak, $R_0$. This value is normalized by the effective minor radius of the last closed magnetic flux surface $\rho_b$, which is considered to be the plasma boundary, i.e

$$\hat{\rho} \triangleq \frac{\rho}{\rho_b}. \tag{10}$$

The output state $y$ in this paper is the safety factor ($q$)-profile. The safety factor is closely related to the magneto-hydrodynamic stability of the plasma, and is defined as $q(t, \hat{\rho}) \triangleq d\Phi/d\Psi$. Using the relation, $\Phi = \pi B_{\phi,0}\rho^2$, the safety factor can be expressed as

$$q(t, \hat{\rho}) = -\frac{d\Phi}{d\Psi} = -\frac{d\Phi}{2\pi d\psi} = -\frac{\frac{\partial\Phi}{\partial\rho}\frac{\partial\rho}{\partial\hat{\rho}}}{2\pi\frac{\partial\psi}{\partial\hat{\rho}}} = -\frac{B_{\phi,0}\rho_b^2\hat{\rho}}{\partial\psi/\partial\hat{\rho}}, \tag{11}$$

where $\psi(t, \hat{\rho})$ is the poloidal stream function, closely related to the poloidal magnetic flux ($\Psi = 2\pi\psi$). The state variable $x$ is chosen as $\theta \triangleq \partial\psi/\partial\hat{\rho}$.

The poloidal flux evolution in a tokamak is governed by the Magnetic Diffusion Equation (MDE) [11,12],

$$\frac{\partial\psi}{\partial t} = \frac{\eta(T_e)}{\mu_0\rho_b^2\hat{F}^2}\frac{1}{\hat\rho}\frac{\partial}{\partial\hat\rho}\left(\hat\rho\hat{F}\hat{G}\hat{H}\frac{\partial\psi}{\partial\hat\rho}\right) + R_0\hat{H}\eta(T_e)\frac{\langle\bar{j}_{ni}\cdot\bar{B}\rangle}{B_{\phi,0}}, \tag{12}$$

where $\eta$ is the plasma resistivity, $T_e$ is the electron temperature, $\mu_0$ is the vacuum permeability, $j_{ni}$ is total current density from noninductive sources, $\bar{B}$ is the magnetic field, $\langle\rangle$ denotes a flux surface average, and $\hat{F}$, $\hat{G}$, and $\hat{H}$ are equilibrium parameters. This equation is closed by control-oriented models for the electron density, electron temperature, plasma resistivity, and non-inductive current drives, first developed in [12] and tailored to NSTX-U in [13]. The goal of these control oriented models is to capture the dominant physics that affect the overall evolution of the system in response to the different actuators.

By rearranging the MDE (12) and combining terms, it is possible to derive a partial differential equation that governs the $\theta$ evolution,

$$\dot\theta = \left[h_{\eta,1}\theta'' + h_{\eta,2}\theta' h_{\eta,3}\right]\tilde{u}_\eta + \sum_{k=1}^{K} h_{nbi,k}\tilde{u}_{nbi,k} + \left[\frac{h_{bs,1}}{\theta} + \frac{h_{bs,2}}{\theta^2}\theta'\right]\tilde{u}_{bs}, \tag{13}$$

where $h_{\eta,1}$, $h_{\eta,2}$, $h_{\eta,3}$, $h_{nbi,k}$, $h_{bs,1}$, and $h_{bs,2}$ are all spatial profiles composed of plasma parameters. The derivation of (13) from the MDE and the definition for the spatial profiles can be found in [14]. Note that the index ($'$) denotes derivatives with respect to $\hat\rho$. The vectors $\tilde{u}_\eta$, $\tilde{u}_{nbi,k}$, and $\tilde{u}_{bs}$ are functions of the inputs,

$$\tilde{u}_\eta(t) \triangleq I_p^{-3/2}P_{tot}^{-3/4}\bar{n}_e^{3/2}\,, \tag{14}$$

$$\tilde{u}_{nbi,k}(t) \triangleq I_p^{-1}P_{tot}^{-1/2}P_{nbi,k}\,, \tag{15}$$

$$\tilde{u}_{bs}(t) \triangleq I_p^{-1/2}P_{tot}^{-1/4}\bar{n}_e^{3/2}, \tag{16}$$

where $I_p$ is the plasma current, $\bar{n}_e$ is the line averaged electron density, and $P_{nbi,k}$ is the power of the $k$th neutral beam injector. The variable $P_{tot}$ is the total power, $P_{tot} = P_{ohm} - P_{rad} + \eta_{fus}P_{fus} + \sum_{k=1}^{K}P_{nbi,k}$, where $P_{ohm}$ is the ohmic power, $P_{rad}$ is the radiative power, $P_{fus}$ is the fusion power, and $\eta_{fus}$ is an efficiency constant that captures the effectiveness of the fusion power in heating the plasma. The models used to calculate each of these power contributions are found in [15].

This parabolic partial differential equation has the boundary conditions

$$\theta(t,0) = 0, \qquad \theta(t,1) = -k_{I_p}I_p(t). \tag{17}$$

The variable $k_{I_p}$ is a constant related to the plasma shape, $k_{I_p} = \mu_0R_0/[2\pi\hat{G}(1)\hat{H}(1)]$.

Notice how (13) is organized as the sum of spatial profiles multiplied by scalar values that vary in time. This equation can be reorganized in matrix form. Before this is done, (13) is discretized in space using the finite difference method over $M+2$ nodes between $\hat\rho = 0$ and $\hat\rho = 1$, This transforms the spatial coordinate from a continuous value to a series of discrete points, $\hat\rho \to \hat\rho_m$, $m = 0, 1, \ldots, M+1$. Since the plasma state at the boundaries is rewritten in terms of the plasma state at the interior points by using the boundary conditions, the new state variable is defined as $\boldsymbol{\theta} = \left[\theta_1(t), \ldots\theta_m(t), \ldots\theta_M(t)\right]' \in \mathbb{R}^{M\times1}$. The vector containing all discrete interior spatial nodes is defined as $\hat{\boldsymbol{\rho}} = \left[\hat\rho_1, \ldots\hat\rho_m, \ldots\hat\rho_M\right]' \in \mathbb{R}^{M\times1}$. Each spatial first and second derivative in (13) is approximated using the neighboring values of said quantity. The following definitions are introduced,

$$\chi_m \triangleq \frac{h_{\eta,1}(\hat\rho_m)}{\Delta\hat\rho^2} - \frac{h_{\eta,2}(\hat\rho_m)}{2\Delta\hat\rho}, \tag{18}$$

$$\beta_m \triangleq \frac{h_{\eta,1}(\hat\rho_m)}{\Delta\hat\rho^2} + \frac{h_{\eta,2}(\hat\rho_m)}{2\Delta\hat\rho}, \tag{19}$$

$$\gamma_m \triangleq h_{\eta,3}(\hat\rho_m) - \frac{2h_{\eta,1}(\hat\rho_m)}{\Delta\hat\rho^2}. \tag{20}$$

The discrete form of (13) for each $m$th inner node, $m = 1, 2, \ldots, M$, can now be written as

$$\dot\theta_m = \left[\chi_m\theta_{m-1} + \gamma_m\theta_m + \beta_m\theta_{m+1}\right]\tilde{u}_\eta(t) + \sum_{k=1}^{K}h_{nbi,k}^m\tilde{u}_{nbi,k}(t) + \left[\frac{h_{bs,1}^m}{\theta_m} - \frac{h_{bs,2}^m}{\theta_m^2}\left(\frac{\theta_{m+1}-\theta_{m-1}}{2\Delta\hat\rho}\right)\right]\tilde{u}_{bs}(t). \tag{21}$$

The evolution of the discretized $\boldsymbol{\theta}$ can now be represented as a matrix equation,

$$\dot{\boldsymbol{\theta}} = \boldsymbol{G}(\boldsymbol{\theta}, I_p)\tilde{u}, \tag{22}$$

where $\boldsymbol{G}$ is a matrix composed of the spatial profiles, $\boldsymbol{G} = [\boldsymbol{G}_{eta}, \boldsymbol{G}_{nbi,1}, \ldots, \boldsymbol{G}_{nbi,K}, \boldsymbol{G}_{bs}]$. Each element in the $\boldsymbol{G}$ matrix is defined as

$$G_\eta^m \triangleq \chi_m\theta_{m-1} + \beta_m\theta_{m+1} + \gamma_m\theta_m, \tag{23}$$

$$G_{nbi,k}^m \triangleq h_{nbi,k}^m, \tag{24}$$

$$G_{bs}^m \triangleq \frac{h_{bs,1}^m}{\theta_m} - \frac{h_{bs,2}^m}{\theta_m^2}\left(\frac{\theta_{m+1}-\theta_{m-1}}{2\Delta\hat\rho}\right). \tag{25}$$

It is important to note that since the state $\boldsymbol{\theta}(t)$ is now a discretized vector, the output must be discretized as well, $q(t,:) \to \boldsymbol{q} = \left[q_1(t), \ldots q_m(t), \ldots q_M\right]' \in \mathbb{R}^{M\times1}$. The output equation for the system is the discretized version of (11),

$$\boldsymbol{q}(t) = -\frac{B_{\phi,0}\,\rho_b^2\,\hat{\boldsymbol{\rho}}}{\boldsymbol{\theta}} \triangleq g(\boldsymbol{\theta}), \tag{26}$$

## 4. Optimization problem formulation

The cost function is designed to minimize the difference between the output variable $\boldsymbol{q}$ and the target output variable, $\boldsymbol{q}^*$, at various discrete times throughout the tokamak shot. It is given by

$$J(\boldsymbol{u}) = \sum_{j=1}^{N_{\hat t}} w_t(\hat t_j)\underbrace{\sum_{m=1}^{M} w_{\hat\rho}(\hat\rho_m)(q_m(\hat t_j) - q_m^*(\hat t_j))^2}_{e_q(\hat t_j)}\,. \tag{27}$$

The variable $\hat t_j$, for $j = 1, 2, \ldots, \hat T$ is the time at which the $q$-profile is compared to the target. There are two user-defined weights introduced in this cost function, a time weight $w_t$ and a spatial weight $w_{\hat\rho}$.

The inputs to the optimization problem are the powers of the neutral beam injectors $P_{nbi,k}$, as well as the plasma current $I_p$, at specific times throughout the simulation,

$$\boldsymbol{u} = \left[\boldsymbol{P}_{nbi,1}, \quad \ldots \quad \boldsymbol{P}_{nbi,K}, \quad \boldsymbol{I}_p\right]', \tag{28}$$

$$\boldsymbol{P}_{nbi,k} = \left[P_{nbi,k}(\bar t_1) \quad \ldots \quad P_{nbi,k}(\bar t_{\bar T})\right]', \tag{29}$$

$$\boldsymbol{I}_p = \left[I_p(\bar t_1) \quad \ldots \quad I_p(\bar t_{\bar T})\right]'. \tag{30}$$

where $\bar t_i$ are the discrete control times defined in (3)–(4). As stated in Section 2, the values of the actuator values at any time ($P_{nbi,k}(t)$, $I_p(t)$) are recovered from the input parameters using linear interpolation between the two bounding times in the input time vector $\bar{\boldsymbol{t}}$. For any such time $t_s$, such that $\bar t_i < t_s < \bar t_{i+1}$, the value of input $u_n$ at $t_s$ is determined using the equation

$$u_n(t_s) = \frac{u_n(\bar t_{i+1}) - u_n(\bar t_i)}{\bar t_{i+1} - \bar t_i}(t_s - \bar t_i) + u_n(\bar t_i). \tag{31}$$

This method can be used to recover the NBI powers and plasma current at any time from the input vector. In particular, the plasma current can be expressed as

$$I_p = p_1(\boldsymbol{u}). \tag{32}$$

By combining the interpolation Eq. (31), with (14)–(16), it is possible to calculate $\tilde u$ from the inputs $\boldsymbol{u}$,

$$\tilde u = p_2(\boldsymbol{u}). \tag{33}$$

By substituting the cost function (27), the state equation (22), the input transformation Eqs. (32) and (33), and the output equation (26) into (5)–(7), it is possible to define the optimization problem as

$$\min_{\boldsymbol{u}} J(\boldsymbol{u}) \tag{34}$$

$$\text{such that :}\quad \dot{\boldsymbol{\theta}} = \boldsymbol{G}(\boldsymbol{\theta}, I_p)\tilde{u}(t), \tag{35}$$

$$I_p = p_1(\boldsymbol{u}), \tag{36}$$

$$\tilde{u} = p_2(\boldsymbol{u}), \tag{37}$$

$$\boldsymbol{q} = g(\boldsymbol{\theta}), \tag{38}$$

$$\hat{\boldsymbol{u}} \leq \boldsymbol{u} \leq \bar{\boldsymbol{u}}. \tag{39}$$

Note that (39) determines the range constraints on the input, with $\hat{\boldsymbol{u}}$ being the minimum allowable input and $\bar{\boldsymbol{u}}$ being the maximum allowable input. There are no state constraints in this particular problem.

## 5. Taking analytical gradients of the cost function

The gradients of the cost function are decomposed into a series of partial derivatives,

$$\underbrace{\frac{\partial J}{\partial \boldsymbol{u}}}_{\in\mathbb{R}^{1\times N*\bar{T}}} = \underbrace{\left(\frac{\partial J}{\partial \boldsymbol{e}_q}\right)}_{\in\mathbb{R}^{1\times\hat{T}}} \underbrace{\left(\frac{\partial \boldsymbol{e}_q}{\partial \boldsymbol{q}(t)}\right)}_{\in\mathbb{R}^{\hat{T}\times M}} \underbrace{\left(\frac{\partial \boldsymbol{q}(t)}{\partial \boldsymbol{\theta}(t)}\right)}_{\in\mathbb{R}^{M\times M}} \underbrace{\left(\frac{\partial \boldsymbol{\theta}(t)}{\partial \boldsymbol{u}}\right)}_{\in\mathbb{R}^{M\times N*\bar{T}}}. \tag{40}$$

Note that $N * \bar{T}$ is the size of the input vector $\boldsymbol{u}$. Using the equation for the cost function (27) and the plasma dynamics (22), (26), it is possible to solve for each of these partial derivatives. Each partial derivative with the exception of $(\partial\boldsymbol{\theta}/\partial\boldsymbol{u})$ reduces to an algebraic equation, which can be easily calculated. This final partial derivative, however, involves taking the derivative of (22) with respect to the input variables and integrating it over time,

$$\left(\frac{\partial \boldsymbol{\theta}(t)}{\partial \boldsymbol{u}}\right) = \frac{\partial \boldsymbol{\theta}(t_0)}{\partial \boldsymbol{u}} + \int_{t_0}^{t} \frac{\partial \dot{\boldsymbol{\theta}}}{\partial \boldsymbol{u}} dt. \tag{41}$$

Solving (41) requires repeated calculations of an ordinary differential equation $\partial\dot{\boldsymbol{\theta}}/\partial\boldsymbol{u}$, and therefore necessitates the bulk of the computation.

### 5.1. Calculating the derivative of the state equation

Starting with the matrix form of the state equation, (22), the gradient of $\dot{\boldsymbol{\theta}}$ with respect to the input $\boldsymbol{u}$ can be expressed as

$$\underbrace{\frac{\partial \dot{\boldsymbol{\theta}}}{\partial \boldsymbol{u}}}_{\in\mathbb{R}^{M\times N*\bar{T}}} = \underbrace{\left(\frac{\partial(\mathbf{G}\tilde{u})}{\partial u(t)}\right)}_{\in\mathbb{R}^{M\times N}} \underbrace{\left(\frac{\partial u(t)}{\partial \boldsymbol{u}}\right)}_{\in\mathbb{R}^{N\times N*\bar{T}}} + \underbrace{\left(\frac{\partial(\mathbf{G}\tilde{u})}{\partial \boldsymbol{\theta}(t)}\right)}_{\in\mathbb{R}^{M\times M}} \underbrace{\left(\frac{\partial \boldsymbol{\theta}(t)}{\partial \boldsymbol{u}}\right)}_{\in\mathbb{R}^{(M)\times N*\bar{T}}}. \tag{42}$$

Since $\partial\dot{\boldsymbol{\theta}}/\partial\bar{\boldsymbol{u}} = \partial(\partial\boldsymbol{\theta}/\partial\boldsymbol{u})/\partial t$, (42) can be considered an ordinary differential equation of $(\partial\boldsymbol{\theta}/\partial\boldsymbol{u})$. The derivatives in (42) are decomposed further using the chain rule,

$$\left(\frac{\partial(\boldsymbol{G}\tilde{u})}{\partial u(t)}\right) = \mathbf{B}_1 + \boldsymbol{G}\left(\frac{\partial\tilde{u}}{\partial u(t)}\right), \tag{43}$$

$$\left(\frac{\partial(\boldsymbol{G}\tilde{u})}{\partial \theta(t)}\right) = \mathbf{B}_2 + \boldsymbol{G}\left(\frac{\partial\tilde{u}}{\partial \theta(t)}\right). \tag{44}$$

The matrices $\mathbf{B}_1 \in \mathbb{R}^{M\times N}$ and $\mathbf{B}_2 \in \mathbb{R}^{M\times M}$ are both time dependent derivative matrices. The $n$th column of $\mathbf{B}_1$ is $\frac{\partial \boldsymbol{G}}{\partial u_n(t)}\tilde{u}$, and the $m$th column of $\mathbf{B}_2$ is $\frac{\partial \boldsymbol{G}}{\partial \theta_m}\tilde{u}$. The remaining gradient matrices are computed using the system dynamics defined in Section 3. Since the vector $\tilde{u}$ contains $P_{tot}$ terms, seen in (14)–(16), it is partially dependent on the plasma dynamics. However, using the power calculation models in [15], none of the power contributions to $P_{tot}$ depend on the magnetic flux, and therefore $\partial\tilde{u}/\partial\theta = 0$. The partial derivative $(\partial u(t)/\partial\boldsymbol{u})$ relates the actuator values at a specific time to the input parameters at specific chosen time points, and is calculated from the linear interpolation Eq. (31).

## 6. Simulation testing

Simulation testing was conducted for this scenario optimizer using the Control Oriented Transport SIMulator (COTSIM) code. The simulation was modeled after the TRANSP run 142301K91, which adopted the NSTX experimental shot 142301 for NSTX-U parameters. The optimizer with analytical gradient calculation was compared with an optimizer with traditional numerical gradient calculation. The target safety factor profiles were created by choosing semi-random, realistic actuator values and running a simulation. The target actuator waveforms were created with the same time parameterization via linear interpolation as the optimizer. Therefore, if the number of time parameters for the optimizer was equal to the number of time parameters used for the creation of the target, the optimizer could match the target exactly. There were two optimization tests conducted. For the first, Test 1, the optimizer time parameters matched those of the target, and therefore the minimized cost function was expected to approach 0. The second test, Test 2, was conducted when the optimizer has a different time parameterization and therefore could not perfectly recreate the target actuator trajectories. In this case the optimizer was expected to converge to some value greater than 0.

### 6.1. Simulation setup

There are two neutral beam injectors on NSTX-U, each with three tangency radii for the beamline to create a total of 6 neutral beam current sources. Four of these sources along with the plasma current were chosen as potential actuators for the simulation testing. Therefore the input vector was composed of

$$\boldsymbol{u} = \begin{bmatrix}\bar{\boldsymbol{P}}_{nbi,1}, & \boldsymbol{P}_{nbi,2}, & \boldsymbol{P}_{nbi,3}, & \boldsymbol{P}_{nbi,4}, & \boldsymbol{I}_p\end{bmatrix}'. \tag{45}$$

The simulation was run between 0.5 and 4 seconds. The state $\theta$ and output $\boldsymbol{q}$ were discretized using 41 spatial nodes, and the number of interior nodes $M = 39$. The initial values of the inputs, $u(0)$, were considered fixed.

For Test 1, the inputs were parameterized at six equidistant control times, and for Test 2, five equidistant control times. Therefore, the time parameterization vectors for Test 1 and Test 2 were

$$\bar{t}_1 = \begin{bmatrix}0.6, & 1.28, & 1.96, & 2.64, & 3.32, & 4\end{bmatrix}, \tag{46}$$

$$\bar{t}_2 = \begin{bmatrix}0.6, & 1.45, & 2.3, & 3.15, & 4\end{bmatrix}. \tag{47}$$

Five target times were chosen where the safety factor profile would be measured against a target profile. For both tests these target times were

$$\hat{\boldsymbol{t}} = \begin{bmatrix}2, & 2.5, & 3, & 3.5, & 4\end{bmatrix}'. \tag{48}$$

The weighting vectors for each simulation test were

$$\boldsymbol{w}_t = \begin{bmatrix}1, & 1, & 1, & 2, & 4\end{bmatrix}, \tag{49}$$

$$\boldsymbol{w}_{\hat{\rho}} = \begin{bmatrix}\mathbf{15}, & \mathbf{10}, & \mathbf{5}, & \mathbf{1}\end{bmatrix}. \tag{50}$$

Note that the time weight vector $\boldsymbol{w}_t$ corresponds to each target time $\hat{\boldsymbol{t}}$. The effect of this weighting vector is that later times contributed more to the cost function $J$. The spatial weight vector was composed of four constant value vectors. The vector $\mathbf{15} \triangleq [15, \ldots\ 15] \in \mathbb{R}^{1\times12}$, and similarly, $\mathbf{10} \triangleq [10, \ldots 10] \in \mathbb{R}^{1\times9}$, $\mathbf{5} \triangleq [5, \ldots 5] \in \mathbb{R}^{1\times15}$, and $\mathbf{1} \triangleq [1, \ldots\ 1] \in \mathbb{R}^{1\times3}$. Note that the combined lengths of these vectors equals 39, the number of interior spatial nodes. Shown by (50), the $q$ values closer to the magnetic axis are weighted significantly higher.

The magnitude constraints on the inputs were as follows

$$0 \leq P_{nbi,k}(t) \leq 2.1, \tag{51}$$

$$0.6 \leq I_p(t) \leq 2.0, \tag{52}$$

These constraints were derived from the actuator limitations of NSTX-U [16].

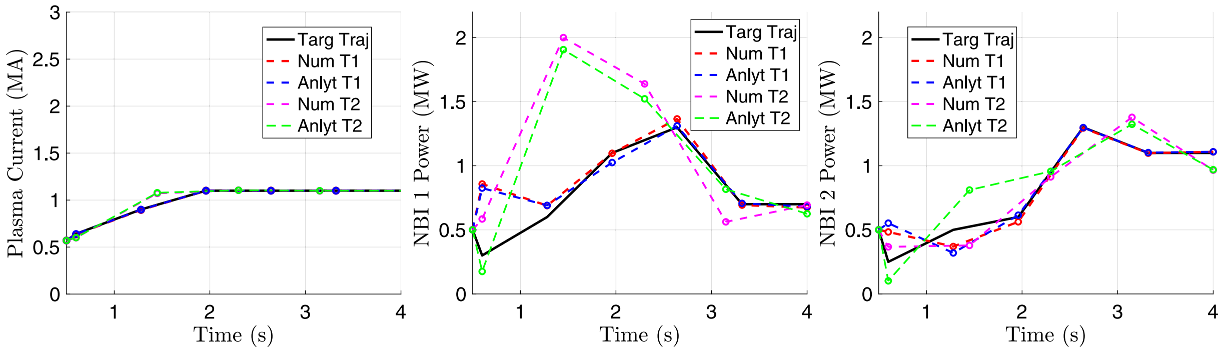

**Fig. 1.** From left to right - (i) Plasma Current Trajectory, (ii) NBI 1 Power Trajectory, (iii) NBI 2 Power Trajectory.

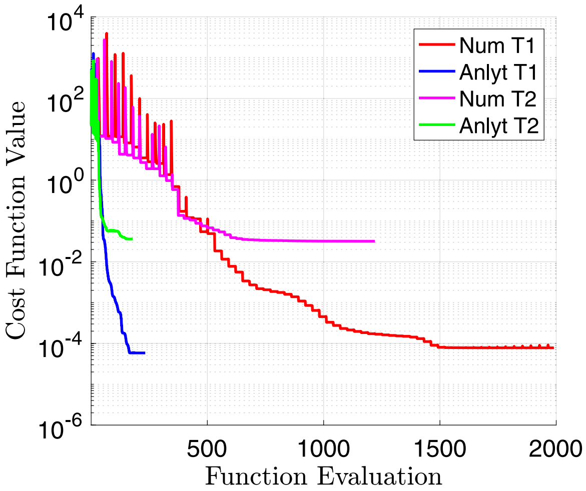

**Fig. 2.** Cost Function of each Iteration for Test 1 and Test 2.

**Table 1**
Simulation results.

| Test | Grad | Initial $J$ | Final $J$ | Run time |
|---|---|---|---|---|
| Test 1 | Num | 2.31e1 | 7.79e−5 | 18:09 |
| | Anlyt | 2.31e1 | 5.84e−5 | 05:42 |
| Test 2 | Num | 2.08e1 | 3.17e−2 | 9:38 |
| | Anlyt | 2.08e1 | 3.61e−2 | 03:26 |

### *6.2. Simulation testing results*

Table 1 shows the simulation results for Tests 1 and 2. For Test 1, the optimizer with analytic gradient calculation reached a lower cost function value in a shorter time frame, being approximately 3 times faster than the numerical optimizer. For Test 2, the two optimizers tested reached comparable optimal $J$ values, and once again the analytic optimizer was significantly faster. Fig. 1 compares a selection of the optimized input parameters for Test 1 and Test 2. Fig. 2 shows the cost function evaluation with each iteration of the SQP method. Fig. 3 compares the output profiles at specific times, and Fig. 4 compares the output trajectories at specific $\hat{\rho}$ values. For Test 1, the input waveforms generated by each optimizer were similar to the ones used to create the target profile. This demonstrates that each optimizer was converging on the optimal solution and thus working as intended. For Test 2, the optimizer waveforms had slightly more deviation. However, Figs. 3 and 4 show that both optimizers in each case were able to get almost perfect matches to the target safety factor profiles. This demonstrates that each optimizer was able to satisfy the control objective. However, the analytic solver did so in a much shorter time frame, and therefore demonstrates an improvement over the numerical optimizer.

## 7. Conclusion

A model-based optimizer for the safety factor profile with analytical gradient calculation based on a control-oriented model has been created for NSTX-U. The optimizer uses sequential quadratic programming to minimize a user-defined cost function subject to various constraints. The optimizer has been improved upon previous attempts at scenario optimization by introducing analytical gradient calculation of the cost function with respect to the input parameters, thus significantly reducing the computation time. This optimizer was tested in an NSTX-U simulation and compared to an optimizer that used numerical gradients. The results showed that the incorporation of analytical gradients into the optimization algorithm significantly reduces the computational time. Future extensions could involve expanding the plasma response model to predict other plasma parameters such as the rotation profile, internal inductance, or normalized beta in order to achieve a broader scenario optimization by using the fast optimization technique proposed in this work. Additionally, the control-oriented models based

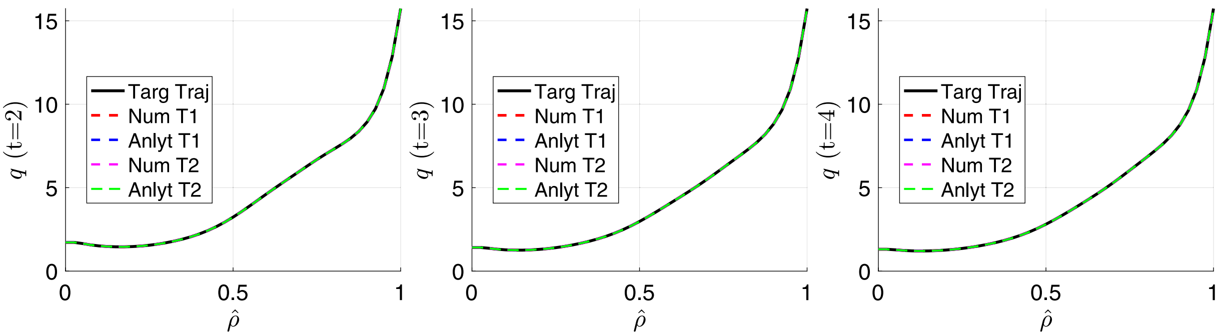

**Fig. 3.** From left to right - (i) $q(t = 2, \cdot)$, (ii) $q(t = 3, \cdot)$, (iii) $q(t = 4, \cdot)$.

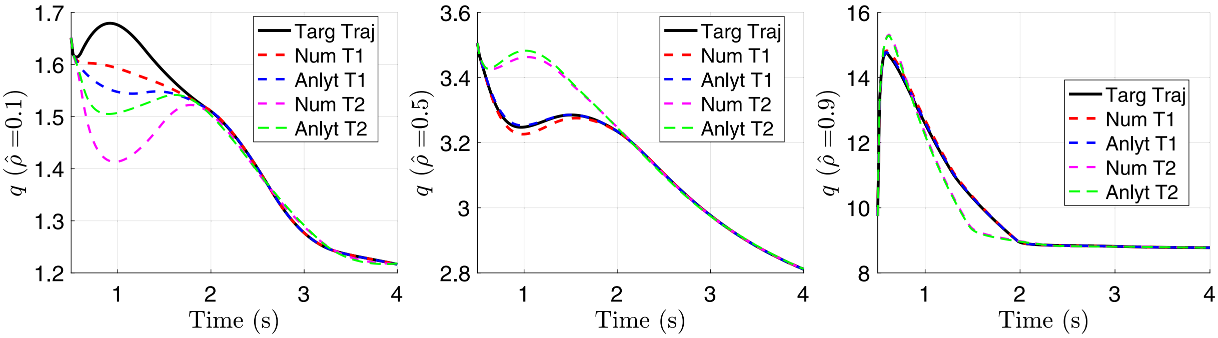

**Fig. 4.** From left to right - (i) $q(\cdot, \hat{\rho} = 0.1)$, (ii) $q(\cdot, \hat{\rho} = 0.5)$, (iii) $q(\cdot, \hat{\rho} = 0.9)$.

on empirical scalings could be replaced by transport equations and machine-learning-based surrogate models for sources and transport.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Data availability

The authors do not have permission to share data.

## Acknowledgments

This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Fusion Energy under Award Number DE-SC0021385.

## References

[1] Y Ou, C Xu, E Schuster, T C Luce, J R Ferron, M L Walker, D A Humphreys, Design and simulation of extremum-seeking open-loop optimal control of current profile in the DIII-D Tokamak, Plasma Phys. Control. Fusion 50 (11) (2008) 115001.

[2] Justin E. Barton, Wenyu Shi, Eugenio Schuster, Tim C. Luce, John R. Ferron, Michael L. Walker, David A. Humphreys, Francesca Turco, Robert D. Johnson, Ben G. Penaflor, Nonlinear physics-model-based actuator trajectory optimization for advanced scenario planning in the DIII-D Tokamak, in: 19th IFAC World Congress, Vol. 47, no. 3, 2014, pp. 671–676.

[3] Hexiang Wang, Eugenio Schuster, Tariq Rafiq, Arnold Kritz, Siye Ding, Model-based optimal scenario planning in EAST, in: Proceedings of the 29th Symposium on Fusion Technology, Vol. 123, SOFT-29 Prague, Czech Republic, September 5-9, 2016, 2017, pp. 569–573.

[4] Zeki Okan Ilhan, Model-Based Optimization and Feedback Control of the Current Density Profile Evolution in NSTX-U, Lehigh University, 2016.

[5] F. Felici, O. Sauter, Non-linear model-based optimization of actuator trajectories for Tokamak plasma profile control, Plasma Phys. Control. Fusion 54 (2) (2012) 025002.

[6] D. Moreau, J.F. Artaud, J.R. Ferron, C.T. Holcomb, D.A. Humphreys, F. Liu, T.C. Luce, J.M. Park, R. Prater, F. Turco, M.L. Walker, Combined magnetic and kinetic control of advanced Tokamak steady state scenarios based on semi-empirical modelling, Nucl. Fusion 55 (6) (2015) 063011.

[7] T. Ravensbergen, P.C. de Vries, F. Felici, T.C. Blanken, R. Nouailletas, L. Zabeo, Density control in ITER: An iterative learning control and robust control approach, Nucl. Fusion 58 (1) (2017) 016048.

[8] E. Schuster, W.P. Wehner, J.E. Barton, M.D. Boyer, T.C. Luce, J.R. Ferron, C.T. Holcomb, M.L. Walker, D.A. Humphreys, W.M. Solomon, B.G. Penaflor, R.D. Johnson, Enhanced reproducibility of L-mode plasma discharges via physics-model-basedq-profile feedback control in DIII-D, Nucl. Fusion 57 (11) (2017) 116026.

[9] Jasbir Arora, Introduction to Optimum Design, Elsevier, 2004.

[10] C. Xu, Y. Ou, J. Dalessio, E. Schuster, T.C. Luce, J.R. Ferron, M.L. Walker, D.A. Humphreys, Ramp-up-phase current-profile control of Tokamak plasmas via nonlinear programming, IEEE Trans. Plasma Sci. 38 (2) (2010) 163–173.

[11] F.L. Hinton, R.D. Hazeltine, Theory of plasma transport in toroidal confinement systems, Rev. Modern Phys. 48 (2) (1976) 239–308.

[12] Y. Ou, T.C. Luce, E. Schuster, J.R. Ferron, M.L. Walker, C. Xu, D.A. Humphreys, Towards model-based current profile control at DIII-D, in: Proceedings of the 24th Symposium on Fusion Technology, Vol. 82, no. 5, 2007, pp. 1153–1160.

[13] Zeki O. Ilhan, Justin E. Barton, Eugenio Schuster, David A. Gates, Stefan P. Gerhardt, Jonathan E. Menard, Physics-based control-oriented modeling of the current density profile evolution in NSTX-Upgrade, in: Proceedings of the 29th Symposium on Fusion Technology, Vol. 123, SOFT-29 Prague, Czech Republic, September 5-9, 2016, 2017, pp. 564–568.

[14] Andres Pajares, Eugenio Schuster, Current profile and normalized beta control via feedback linearization and Lyapunov techniques, Nucl. Fusion 61 (3) (2021) 036006.

[15] Justin E Barton, Karim Besseghir, Jo Lister, Eugenio Schuster, Physics-based control-oriented modeling and robust feedback control of the plasma safety factor profile and stored energy dynamics in ITER, Plasma Phys. Control. Fusion 57 (11) (2015) 115003.

[16] S.P. Gerhardt, R. Andre, J.E. Menard, Exploration of the equilibrium operating space for NSTX-upgrade, Nucl. Fusion 52 (8) (2012) 083020.